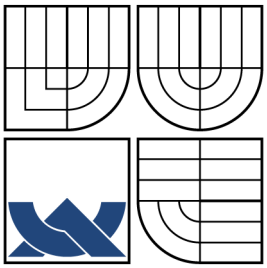
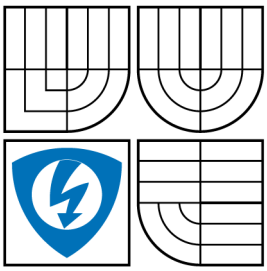


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

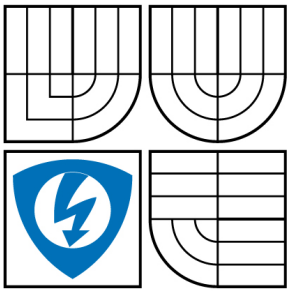
# SPRÁVA APLIKACE APACHE WEBOVÝM ROZHRANÍM
MANAGEMENT OF APACHE APLICATION WITH WEB GUI

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — ROMAN HOŠEK
AUTHOR

VEDOUCÍ PRÁCE — ING. FILIP JANOVIČ
SUPERVISOR

BRNO 2008



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky a komunikačních technologií**

**Ústav telekomunikací**

# Bakalářská práce

bakalářský studijní obor
**Teleinformatika**

| | | | |
|---|---|---|---|
| ***Student:*** | Hošek Roman | ***ID:*** | 78458 |
| ***Ročník:*** | 3 | ***Akademický rok:*** | 2007/2008 |

**NÁZEV TÉMATU:**

## Správa aplikace apache webovým rozhraním

**POKYNY PRO VYPRACOVÁNÍ:**

Návrh vlastní web - aplikace, pro správu kompletního systému Apache nainstalovaného na servru s operačním systémem Debian. Je potřeba spravovat uživatelsky nenáročným prostředím všechny složky aplikace apache.

**DOPORUČENÁ LITERATURA:**

[1] CHARLES, Aulds. Linux administrace serveru Apache. 1st edition. [s.l.] : [s.n.], 2003.
[2] ANDI, Gutmans. Mistrovství v PHP5. 1st edition. [s.l.] : [s.n.], 2007.

| | | | |
|---|---|---|---|
| ***Termín zadání:*** | 11.2.2008 | ***Termín odevzdání:*** | 4.6.2008 |

***Vedoucí práce:*** Ing. Filip Janovič

**prof. Ing. Kamil Vrba, CSc.**
*předseda oborové rady*

**UPOZORNĚNÍ:**

# LICENČNÍ SMLOUVA
# POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

**1. Pan/paní**

Jméno a příjmení: Roman Hošek
Bytem: Moravská 254, 57001, Litomyšl - Litomyšl-Město
Narozen/a (datum a místo): 14.12.1985, Litomyšl

(dále jen "autor")

a

**2. Vysoké učení technické v Brně**

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 244/53, 60200 Brno 2
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Ing. Kamil Vrba, CSc.

(dále jen "nabyvatel")

## Článek 1
## Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   - ☐ disertační práce
   - ☐ diplomová práce
   - ☑ bakalářská práce

   jiná práce, jejíž druh je specifikován jako .....

(dále jen VŠKP nebo dílo)

Název VŠKP: Správa aplikace apache webovým rozhraním
Vedoucí/školitel VŠKP: Ing. Filip Janovič
Ústav: Ústav telekomunikací
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

- ☑ tištěné formě - počet exemplářů 1
- ☑ elektronické formě - počet exemplářů 1

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

**Článek 2**

**Udělení licenčního oprávnění**

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizovaní výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
   - ☑ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

**Článek 3**

**Závěrečná ustanovení**

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

..... .....

Nabyvatel Autor

## ABSTRAKT

Tato práce se zabývá návrhem a realizací webové aplikace pro konfiguraci Apache serveru. Rozebírá problematiku konfigurace a jejích možných řešení. Pro výslednou aplikaci byl zvolen skriptovací jazyk PHP, s ukládáním dat do MySQL databáze. Jako hostitelský systém slouží linuxová distribuce Debian. Práce obsahuje popis vlastností použitých součástí a je obzvláště zaměřena na problematiku virtuálních hostitelů. Součástí je i návod na instalaci a obsluhu. Aplikace vadmin zajišťuje snadné vytváření a nastavování virtuálních hostitelů pouze pomocí webového prohlížeče.

## KLÍČOVÁ SLOVA

apache, virtual hosting, Debian, PHP, SQL, správa, web

## ABSTRACT

This bachelor's thesis deals with a design and implementation of a web application aimed at Apache server configuration. It analyses the question of configuration and its possible solutions. The PHP scripting language with the data storage in MySQL database was chosen for the final application. Linux distribution Debian serves as the host system. The thesis contains descriptions of applied components features and it is particularly focused on the question of virtual hosts. It also includes an installation and application manipulation guide . The vadmin system maintains an easy creation of virtual hosts adjustment just by the help of a web browser.

## KEYWORDS

apache, virtual hosting, Debian, PHP, SQL, management, web

HOŠEK R. *Správa aplikace apache webovým rozhraním. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 35 s. Vedoucí bakalářské práce Ing. Filip Janovič.*

# PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Správa aplikace apache webovým rozhraním“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ..... .....

(podpis autora)

# OBSAH

# SEZNAM OBRÁZKŮ

# ÚVOD

Tato práce se věnuje oblasti konfigurace apache serveru přes grafické webové rozhraní. Na základě požadavků byla vytvořena webová aplikace. Aplikace zajišťuje pohodlný přístup k nastavení a umožňuje měnit konfiguraci serveru, modulů a uživatelských účtů on-line. Celá práce je realizována pomocí scriptovacího jazyka PHP a MySQL databáze, kam se ukládají informace o uživatelích, konfiguraci a serverech. Součástí této práce je i návod na instalaci a provoz této webové aplikace.

# 1 APACHE SERVER

## 1.1 Konfigurace

Kompletní nastavení apache serveru je uloženo v konfiguračních souborech s koncovkou *conf*. Ve verzi 1.3 se jednalo o soubor *httpd.conf*, v němž bylo uložené veškeré nastavení včetně virtuálních hostů. Verze 2 a vyšší ukládají nastavení do souboru *apache2.conf*, na jehož konci je proveden include *httpd.conf*, který je však v těchto verzích prázdný a slouží pro zachování kompatibility. Vlastní nastavení virtuálních hostů se také změnilo. Můžeme je definovat klasicky v *httpd.conf* a *apache2.conf*, nebo nově i vytvořením konfiguračního souboru v includované složce. Na konci *apache2.conf* je pak umístěna includovací direktiva pro jakékoliv konfigurační soubory v určité složce. Po instalaci v rootu serveru (obvykle */etc/apache2/*) najdeme dvě takové složky, *sites-avaible* a *sites-enabled*. Jak už je zřejmé z názvu, ve složce *sites-avaible* jsou konfigurační soubory všech volitelných hostů a pomocí symbolických odkazů se potřebné konfigurace linkují do složky *sites-enabled*. V balíčku apache můžeme využít jednoduché programy `a2ensite` - zapíná sítě a `a2dissite` - vypíná sítě.

Konfigurační soubor obsahuje výchozí nastavení programu po startu, zbytek tvoří komentáře popisující kontrétní příkaz a jeho možné volby.

```
# PidFile: The file in which the server should record its process
# identification number when it starts.
#
PidFile /var/run/apache2.pid

#
# Timeout: The number of seconds before receives and sends time out.
#
Timeout 300

#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On
```

Obr. 1.1: Ukázka části souboru httpd.conf

Nejdůležitější příkazy jsou:

- **DocumentRoot** - Adresář pro umístění webových stránek
- **ServerRoot** - Adresář, do kterého byl server nainstalován

- **Port** - Nastavení na jakém portu má server naslouchat, standartně port 80
- **BindAddress (Listen)** - Obdoba port, ale zde můžeme definovat i na jakých IP adresách má server naslouchat. Například Listen 127.0.0.1:8080 znamená, že server bude naslouchat na localhostu na portu 8080.
- **ServerName** - Definice názvu serveru, důležité hlavně u virtuálních hostů založených na jméně.
- **ServerAdmin** - Email správce. který se objeví v případě chybové hlášky.
- **StartServer** - Udává kolik instancí serveru má být po startu spuštěno
- **MinShareServer** - Udává kolik kopií serveru má být minimalne spuštěno
- **MaxShareServer** - Udává kolik kopií serveru může být maximálně spuštěno
- **ServerType** - Udává, jak má server běžet,buď jako standalone = server běží stále, nebo může být spouštěn pomocí démona inet.
- **ErrorLog** - Cesta k souboru pro logování chyb serveru, pokud je zapsána relativně, vztahuje se k rootu serveru.
- **CustomLog** - Cesta k logovacímu souboru, druhým parametrem je druh událostí.
- **LogFormat** - Ovlivňuje typ zápis do logu. Například LogFormat "%User-agenti" agent.
- **PidFile** - Soubor, do kterého se ukládá PID (identifikační číslo proceru)

Dále lze použít hlavní direktivy, jako ServerAdmin atd.
Lze také definovat způsob využívání složek a přístup do nich. Zápis vypadá například takto:

```
<Directory "/htdocs">
    Options Indexes FollowSymLinks Includes
    AllowOverride None
    order allow,deny
    allow from all
</Directory>
```

Kde možné volby Options jsou:

- **None** - Nic není povoleno.

- **All** - Je povoleno vše kromě MultiViews.
- **Indexes** - Pokud není v adresáři nalezen DirectoryIndex, vytvoří seznam souborů v daném adresáři.
- **Includes** - Povoluje všechny vsuvky.
- **IncludesNOEXEC** - Povoluje vsuvky, ale příkazy #include a #exec jsou zakázány.
- **FollowSymLinks** - Povoluje symbolické odkazy.
- **ExecCGI** - Povoluje spouštění CGI skriptů.

**AllowOverride** - určuje, které direktivy se mohou potlačit pro daný adresař. Parametry jsou *AuthConfig, AuthUserFile, FileInfo, Indexes, Limit, Options, All, None.*
**Order allow,deny** určuje v jakém pořadí se zpracovávají allow,deny.
Další možnosti:

<**Files soubor**> ...direktivy... </**Files**>
Umožňuje aplikaci direktiv uvedených v bloku na uvedený soubor.

<**IfModule modul**> ...direktivy... </**IfModule**>
Pokud je modul definován, direktivy se aplikují. Jinak jsou ignorovány.

<**Location URL**>
SetHandler server-status
order deny,allow
deny from all
allow from .feec.vutbr.cz
</**Location**>
Umožňuje aplikaci direktiv uvedených v bloku na uvedené URL.

Základní direktivy pro virtuální hosty:

- <**VirtualHost *:80**> **a** </**VirtualHost**> - Ohraničuje blok virtuálního hosta. Počáteční tag má volitelně dva parametry. První určuje na jaké IP má host naslouchat a druhý určuje na jakém portu.
- **DocumentRoot** - Určuje kořenový adresář hosta.

- **ServerName** - Název serveru, důležité u virtuálních hostů založených na jméně. (Podle ServerName se rozlišuje, který server bude zpracovávat dotaz obsahující shodné jméno v hlavičce požadavku)
- **RewriteLogLevel** - Určuje, jaké akce se mají logovat v rewrite engine, 0 znamené žádné logování, 9 logování všech akcí.
- **RewriteLog** - Soubor do kterého se ukládají rewrite logy.

<**VirtualHost 127.0.0.1**>
ServerAdmin xhosek05@stud.feec.vutbr.cz
DocumentRoot /www/docs/host
ServerName xhosek05.cz
ErrorLog logs/host/error_log
CustomLog /var/log/httpd/host/access_log common
</**VirtualHost**>
Tato direktiva nám umožňuje provozovat virtuální servery.

## 1.2 Možné způsoby konfigurace

### 1.2.1 Ruční editace httpd.conf

Konfigurační soubory apache serveru jsou textové soubory, můžeme je tedy editovat jakýmkoliv textovým editorem. Například vi, nano, Kedit. Neposkytují ale žádnou kontrolu zadávaných údajů ani zvýraznění syntaxe. Proto je práce s nimi nepřehledná, u složitějších, nebo obsáhlých souborů takřka nemožná.

### 1.2.2 Programové GUI

Hlavně kvůli nedostatkům ruční editace pomocí textových editorů začala vznikat programová grafická rozhraní. Může se jednat o nadstavby oblíbených editorů, přidávající zvýraznění syntaxe, nebo o komplexní aplikace. Ty mají pro lepší přehlednost rozhraní rozčleněné na několik částí. Uživatel si pak pouze vybere která část nastavení ho zajímá a nestará se o syntaxi a zápis. Pokročilé aplikace nabízejí i grafické výstupy statistických dat, stavy jednotlivých součástí, rozsáhlou nápovědu, či popis základního nastavení. Nevýhodu je závislost na na operačním systému, případná potřeba specifických knihoven. Některá pokročilá programová GUI umožňují i vzdálené přihlášení, většinou však podmíněné instalací klienstké části programu.

### 1.2.3 Webové GUI

Webové rozhraní řeší většinu těchto problémů. Dovoluje zvýrazňování syntaxe, členění rozhraní. Je nezávislé na operačním systému a umožňuje vzdálené ovládání. Neklade žádné nároky na hostující počítač, je potřeba pouze fungující internetové připojení a prohlížeč. Vše ostatní je zajišťováno samotným serverem. Nevýhodou může být, že aplikace je dostupná odkudkoliv a je tedy potřebné ji dostatečně zabezpečit. To může být provedeno omezením přístupu na určité IP adresy, zaheslovaním, případně kombinací dalších metod.

### 1.2.4 Výběr vhodného řešení

Z předchozích možností se jako nejvýhodnější zdá použít webové rozhraní. Vyhovuje požadavkům pro vzdálený přístup více uživatelů a poskytuje pohodlné rozhraní. Pro specifické požadavky se však většina těchto prostředí nehodí. Proto jsem udělal jednoduché vlastní prostředí. Nezatěžuje uživatele přebytečnými volbami, ale zároveň nabízí administrátorům rozsáhlé možnosti konfigurace.

## 1.3 Virtuální hostitelé

Apache poskytuje užitečný prvek nazvaný virtuální hostitelé. Ten umožňuje hostit více doménových názvů na jednom serveru a používat jednu IP pro všechny virtuální hostitele, nebo přiřadit skupinám hostů rozdílné IP adresy. Vše v široce konfigurovatelné a škálovatelné podobě. Tím lze efektivně rozdělit výpočetní výkon serveru mezi více hostů. Základní vlastnosti virtuálních hostů:

- Virtuální hosté využívají IP adresy, nebo doménové názvy.
- Každý hostitel může mít vlastní DocumentRoot.
- Každý host může mít vlastní protokolování (chybové, přístupů, atd.)
- Možnost sdílení adresářů mezi hosty, pomocí ScriptAlias a Alias.
- Možnost překrývání voleb podle adresářů, nebo hostů.

Rozlišujeme virtuální hosty založené na doménovém jméně, nebo IP adrese.
Virtuální hosté založení na doménovém jméně:

```
NameVirtualHost *
<VirtualHost *>
ServerName www.vadmin.cz
DocumentRoot /www/vadmin
</VirtualHost>
<VirtualHost *>
ServerName www.xhosek05.cz
DocumentRoot /www/xhosek05
</VirtualHost>
```

Server rozpozná z hlavičky zaslané prohlížečem, na jakou doménu přistupujeme a podle toho zvolí hosta.
Virtuální hosté založení na IP adrese:

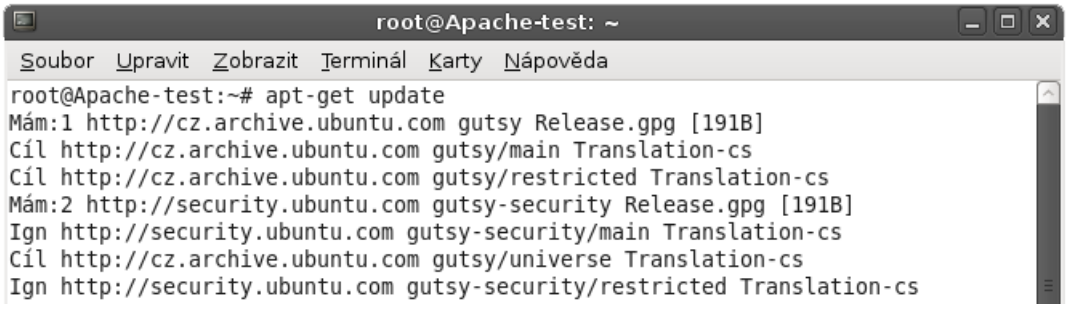
```
NameVirtualHost 127.0.0.1
<VirtualHost 127.0.0.1>
ServerName www.vadmin.cz
ServerPath /vadmin
DocumentRoot /web/vadmin
</VirtualHost>
```

Server může mít více síťových rozhraní s vlastní IP adresou. Podle IP adresy pak volí který host je požadován. Rozlišení může probíhat i pomocí rozdílných portů.

# 2 VLASTNOSTI POUŽITÝCH SOUČÁSTÍ

## 2.1 Linuxová distribuce Ubuntu

Debian je pro svou stabilitu a jednoduchou údržbu velmi oblíbený zejména pro serverové instalace, na pracovních stanicích se však více využívá Ubuntu, komunitní linuxová distribuce založená právě na Debianu. Na jejích oficiálních strákách nalezneme množství variant zdarma ke stažení. Můžeme si například vybrat, zda chceme 32bit nebo 64bit verzi, nebo optimalizaci pro vícejádrové procesory. Pokud si nejsme jisti, zvolíme 32bit verzi, někdy značenou jako x86. Uživatele ale bude spíše zajímat rozdělení na systém určený pro pracovní stanice (označované jako Desktop), nebo systém určený pro serverové nasazení. Hlavní rozdíl je v tom, že serverový systém obvykle nemá v základu naistalované grafické uživatelské rozhraní (X-server), které už z podstaty účelu serveru nepotřebuje a zbytečně by zatěžovalo systémové prostředky, ale v případě potřeby lze volitelně doinstalovat. Velkou výhodou tohoto systému je, jeho velice propracovaný systém balíčků a balíčkovací služba. Ta je dostupná z příkazové řádky pomocí skupiny programů `aptitude` a její nadstavby `Synaptic Package Manager`, používaný v grafickém prostředí `gnome`, nebo `KPackage`, používaný v alternativním grafickém prostředí `KDE`. Tento systém klade menší nároky na znalosti uživatele a spolu s automatickou kontrolou závislostí/kolizí balíčků a velkým množstvím už existujících balíčku na internetu,je velkou výhodu pro začínající uživatele. Existují i jiné balíčkovací systémy, například RPM, vyskytuící se v komerčně šířených distribucích (Red Hat, Mandriva, SUSE), ale v mnoha případech trpící nedostatkem vhodných balíčků.




Obr. 2.1: Ukázka konzole s programem apt a aktualizací repozitářů

Balíčky jsou uloženy na serverech, které se nazývají repozitáře. Tento systém budeme využívat i později pro instalaci vlastního apache serveru.

## 2.2 Skriptovací jazyk PHP

PHP (Hypertext Preprocessor)[1] je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek.

Jeho hlavní výhody jsou:

- Je jednoduché na pochopení
- Syntaxe podobná jazyku C
- Velká podpora rozšiřujících technologií a podpora komunity
- Velké množství ukázkových příkladů, či hotových řešení.
- Velmi dobrá spolupráce s apache.
- Podpora a snadná kominikace s databázemi.
- Je multiplatformní.
- Podporu PHP nalezneme snad na každém hostingu.

Charakteristickými vlastnostmi jsou:

- Jazyk je dynamicky typový, kdy datový typ proměnné se určí podle přiřazené hodnoty.
- Dva druhy porovnávání, klasické == a ===, kdy se porovnává i zda se jedná o stejné datové typy.
- Pole jsou heterogenní, tj můžou obsahovat libovolné informace i v jejích indexech.

PHP interpreter může běžet samostatně v konzoli, nebo jako modul apache serveru. V projektu využívám výhradně PHP jako modul serveru. Skript je zpracováván na straně serveru a k uživateli je zasílán pouze výsledek. Vlastní PHP kód se vlastně začleňuje do struktury HTML. PHP parser pak zpracovává příkazy ohraničené speciálnímy tagy. Těch jsou 4 druhy:

- `<?php a ?>` standardní zápis.
- `<script language=‘‘php’’> </script>` styl zápisu pro editory typu FrontPage.
- `<? nebo <? a ?>` zkrácený zápis (standardně povolen)

[1] Dříve označovaný také jako Personal Home Page.

- <% nebo <% a %> zápis ve stylu ASP (musí být povolen v nastavení, nedoporučuje se používat, nestandardní)

Globální nastavení je uloženo v souboru *php.ini*. Pokud je povoleno, lze nastavení měnit pomocí *php_flag* a *php_value* v definici virtualního hosta, případně soubory *.htaccess*. Od PHP verze 4.2.0 je defaultně zakázáno register globals. To zajišťuje, že u globalních proměnných máme zaručené, že data byla odeslána požadovanou metodou.

## 2.3 SQL databáze

SQL (Structured Query Language). MySQL je relační, multiplatformní a víceuživatelská databáze. Je vyvíjena švédskou firmou MySQL AB. Zjednodušeně lze říci, že je tvořena tabulkami, mezi kterými jsou relační vztahy. Tabulky můžou být různých typů (v projektu je použit nejpoužívanější druh MyISAM.) SQL je deklarativní programovací jazyk, což znamená, že kód nepíšeme samostatně, ale vkládáme jej do jiného procedurálního programovacího jazyka. Jazyk SQL se dělí na dvě hlavní kapitoly na DDL a DML.
**DDL** je zkratka pro Data Definition Language a jedná se o příkazy definující jednotlivé objekty v databázi (tabulky, procedury, view apod.).
**DML** je zkratka pro Data Modification Language a jde o příkazy definující práci s databázovými objekty (výběr dat, modifikaci dat, mazání dat apod.)

Další části jsou:

**SDL** - Storage Definition Language, definuje způsob ukládání tabulek.
**VDL** - View Definition Language, určuje vytváření pohledů (složení tabulek.)

Příkazy pro manipulaci s daty (**DML**).

- **SELECT** – vybírá data z databáze, umožňuje výběr podmnožiny a řazení dat.
- **INSERT** – vkládá do databáze nová data.
- **UPDATE** – mění data v databázi (editace).
- **DELETE** – odstraňuje data (záznamy) z databáze.
- **EXPLAIN PLAN FOR** – speciální příkaz, který zobrazuje postup zpracování SQL příkazu.

- **SHOW** - příkaz, umožňující zobrazit databáze, tabulky nebo jejich definice

Příkazy pro definici dat(**DDL**).

- **CREATE** – vytváření nových objektů.
- **ALTER** – změny existujících objektů.
- **DROP** – odstraňování objektů.

Příkazy pro řízení dat(**DCL**).

- **GRANT** – příkaz pro přidělení oprávnění uživateli k určitým objektům.
- **REVOKE** – příkaz pro odnětí práv uživateli.
- **BEGIN** – zahájení transakce.
- **COMMIT** – potvrzení transakce.
- **ROLLBACK** – zrušení transakce, návrat do původního stavu.

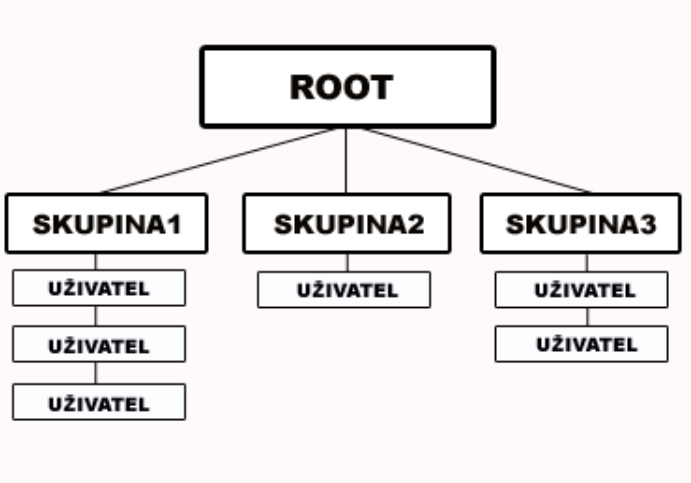
## 2.4 Apache server

Apache je jednoduchý, ale přitom velmi výkonný web server, který je dostupný jak pro platformu Unix/Linux, tak i 32bitové operační systémy Windows. Vývoj Apache začal v roce 1993 v NCSA (National Center for Supercomputing Aplications) na Illinoiské univerzitě. Původní jméno projektu bylo NCSA HTTPd. Administrátoři serverů si pro něj psali různé záplaty (patche), ať už pro rozšíření funkčnosti, zvýšení výkonu, nebo opravení bezpečnostních mezer. O pár let později vznikla Apache Group a zdrojový kod serveru byl kompletně přepsán. Název Apache pochází z anglického „A patchy server.“ Apache podporuje mnoho programovacích jazyků, které nemusí být počítači vůbec nainstalované, ale stačí, pokud se nahraje příslušný modul, který podporuje konkrétní programovací jazyk. Například stačí malý modul PHP, místo celé rozměrné instalace PHP. Je open source a má velice volnou licenci. V současné době je využíván na více než 60% serverů.

# 3 NÁVOD K POUŽITÍ

## 3.1 Vlastnosti aplikace Vadmin

Vadmin je jednoduchá aplikace, umožňující pohodlnou konfiguraci virtuálních hostů a samotného apache serveru. Možnosti nastavení jsou odvozeny od práv uživatele.




Obr. 3.1: Schéma uspořádaní uživatelů aplikace

Root je hlavním správcem cele aplikace a má spřístupněny veškeré volby aplikace. Root je v systému pouze jeden, jen on má možnost vytvářet správce skupin, i bežné uživatele ve všech skupinách. Volí také, jaké možnosti PHP budou dostupné správcům skupin a uživatelům. Správce skupin má práva pouze na konfiguraci účtů uživatelů, spadajících do jeho skupiny. Může měnit uživatelovo nastavení PHP a nastavení adresáře v rozsahu skupiny.

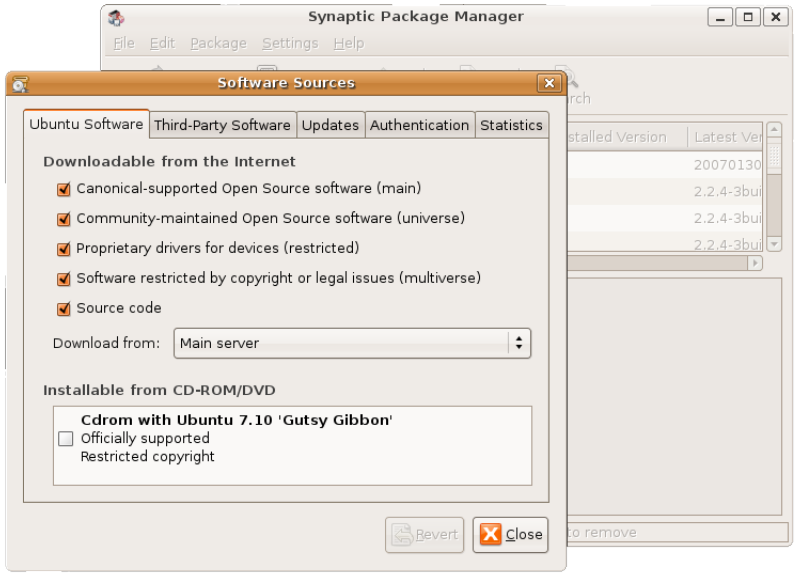
Vadmin pro zlepšení přehlednosti nedefinuje skupiny do hlavního konfiguračního souboru, ale každá skupina má svůj vlastní soubor. Root pak pouze pro aktivaci skupin pouze připíše include souboru skupiny do hlavního souboru, nebo pomocí symbolických odkazů vytvoří odkaz na nastavení ve složce skupin. Správcové skupin se starají o soubor své skupiny a uživatele do ní patřící.

Pokud uživatel edituje soubor a jiný uživatel, patřící do stejné skupiny se k němu pokusí přistoupit, zobrazí se varování. Soubor je tzv. zamknut. Pro řešení situace kdy uživatel při editaci zavře okno prohlížeče, jsou všechny zámky po nastaveném časovém intervalu odstraněny. Vadmin také ve zvoleném čase automaticky zálohuje databázi a nastavení domén do archivu. Lze také nastavit, jak dlouho se mají archivy zálohovat, systém pak automaticky odstraňuje starší archivy.

## 3.2 Instalace potřebných součástí

### 3.2.1 Instalace Apache2, SQL a PHP

Pomocí programu aptitude (případně pomocí jedné z jeho grafických nadstaveb), povolíme získávání balíčků z internetu.




Obr. 3.2: Nastavení prepositiores pro získávání dat z internetu.

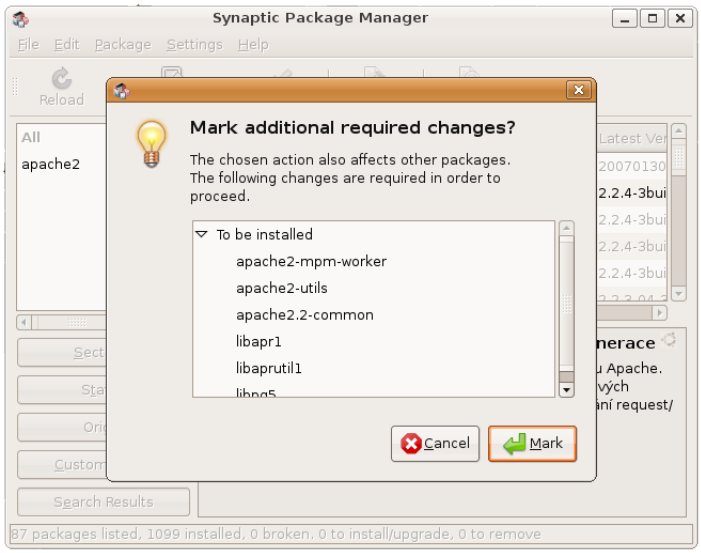
Main jsou oficiální balíčky distributora, universe jsou balíčky vyvíjené komunitou a nadšenci. Po aktualizaci prepositiores můžeme začít vyhledávat potřebné balíky. Pomocí hledat najdeme balíček `apache2`, aptitude automaticky zkontroluje všechny závislé balíčky a nabídne je k instalaci.

Stejným způsobem nainstalujeme `PHP5` a `MySQL-server5` V průběhu instalace budeme požádání o zadání hesla k databázi

Aby se nám s databází lépe pracovalo je dobré naistalovat aplikaci `MyPHPadmin`. Zde budeme při instalaci dotázání, jaký server si přejeme spravovat, vybereme Apache2. Pokud jsme vše udělali dobře, tak po zadání `http://localhost/` do prohlížeče by se měla zobrazit adresářová struktura složky */var/www/*, která je standartně nastavena jako výchozí složka pro internetový obsah serveru.

Pomocí kodu `<?php phpinfo() ?>` si můžeme ověřit funkčnost PHP a jeho nastavení.

Funci MyPHPadmin vyzkoušíme zadáním adresy `http://localhost/myphpadmin`. Pro přihlášení použijeme login root a heslo, co jsme zadávali při instalaci MySQL-serveru.




Obr. 3.3: balíček serveru apache a závislé balíčky.

### 3.2.2 Základní konfigurace

Pomocí libovolného textového editoru si otevřeme soubor *apache2.conf* (standardně */etc/apache2/*). Pozor, abychom mohli v souboru cokoliv měnit musíme mít práva roota! Ke konci souboru přepíšeme nastavení podle obrázku.

Nejdříve provedeme include souboru *vadmin.conf*, to je z duvodu, že Apache2 bere první definici virtuálního hosta jako výchozí pro všechny ostatní. Tímto zajistíme, že v případě přístupu například přímo na IP serveru a nadefinovaných pouze name-based virtualních hostech, se nám zobrazí přihlašovací stránka vadmina. Další include bude už aktivovat skupiny. V tomto souboru je také možné omezit rozsah IP adres, nebo domén, ze kterých bude možné přistupovat do aplikace vadmin.

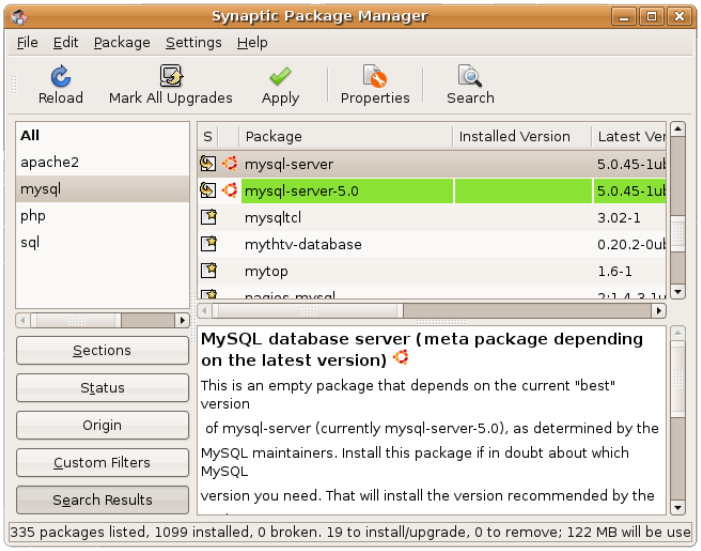
## 3.3 Instalace Vadmina

Nakopírujeme Vadmina do *www* adresáře serveru, například */var/www/vadmin*. Pak pomocí prohlížeče spustíme instalaci.

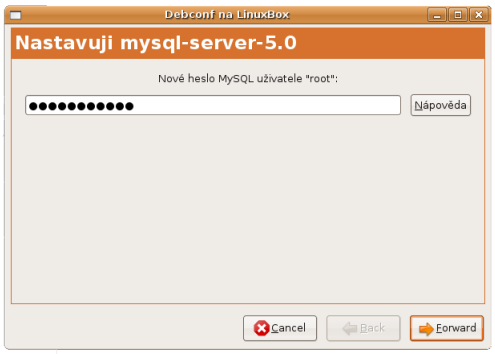
### 3.3.1 Tvorba tabulek

Na SQL serveru vytvoříme například pomocí phpmyadmina databázi. Údaje potřebné k připojení k této databázi zapíšeme do konfiguračního souboru */include/config.php* jako konstanty:

- SQL_HOST - jméno SQL serveru




Obr. 3.4: MySQL-server5.



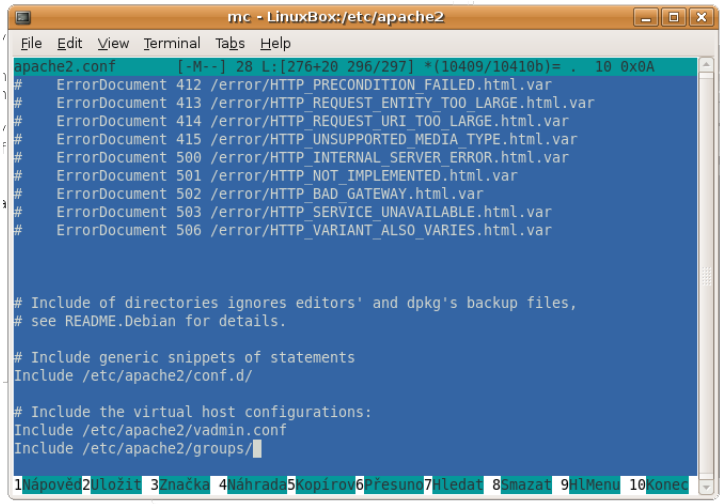

Obr. 3.5: Heslo k SQL.

- SQL_DBNAME - jméno databáze
- SQL_USERNAME - jméno účtu k DB
- SQL_PASSWORD - heslo k účtu k DB

Tvorba tabulek v databázi je automatizovaná. Do adresního řádku prohlížeče napíšeme `http://adresa_serveru/slozka_vadmina/install.php` a pokračujeme podle instrukcí. Tabulky vytvoříme stisknutím tlačítka vytvořit a následně vytvoříme účet roota.

### 3.3.2 Umožnění vzdáleného reloadu

Vzdálený reload nastavení se provádí pomocí jednoduchého shellového skriptu, volaného pomocí PHP skriptu. Z bezpečnostního hlediska nemůže standardně PHP




Obr. 3.6: Přidání vadmin do apache2.conf.

spouštět programy a skripty s právy sudo (superuser do), které jsou však nutné pro jakékoliv operace s apache (start, stop, reload.) PHP má práva uživatele nastaveného v apache2.conf (standardně uživatel www-data, patřící do skupiny www-data.) My tomuto uživateli povolíme provádět sudo příkazy v určité složce. Tato složka by se měla nacházet mimo webový obsah serveru a měla by obsahovat pouze náš skript *vadmin.sh*. Tím zajistíme, že i kdyby útočník dokázal pozměnit PHP, stále je provádění sudo příkazu omezeno pouze na danou složku, která není veřejně přístupná a nelze tedy do ní cokoliv uploadovat, bez úplného přístupu k serveru.

Kod shellového skriptu vadmin.sh

```
#!/bin/sh sudo /etc/init.d/apache2 reload exit 0
```

Povolení sudo příkazu pro uživatele www-data se provede pomocí příkazu `visudo`. Tím se otevře soubor */etc/sudoers*, kde jsou definovaní uživatelé s právy superuživatelů. Soubor se standartně otevře v editoru vi (který jej i zamkne a zajistí výhradní přístup, aby soubor byl v jednom čase editován pouze jedním uživatelem.) Pokud nám vi nevyhovuje, můžeme příkazem `VISUAL=náš_oblíbený_editor` nastavit jako výchozí editor jakýkoliv jiný. Je však důrazně doporučeno používat vi z důvodu správného zalamování řádků. Najdeme pasáž # User privilege specification a upravíme ji podle ukázky:

```
# User privilege specification
root ALL=(ALL) ALL
www-data ALL=(ALL)NOPASSWD: /usr/vadmin/
```

### 3.3.3 Nastavení

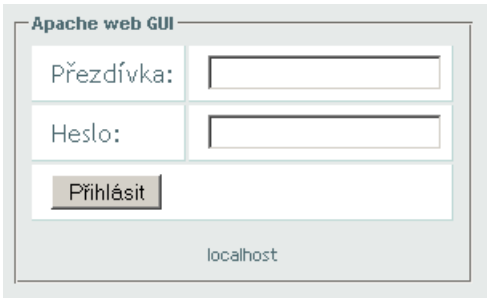
Hlavní nastavení je uloženo v souboru */include/config.php*. Doplníme následující konstanty:

- **BACKUP** - cesta k adresáři pro automatickou zálohu.
- **GROUPS** - cesta k adresáři s konfigurací skupin.
- **DAYS** - kolik dní se archivují zálohy.
- **SECURITY** - cesta k adresáři, kam se ukládají zámky souborů.
- **MINUTES** - po kolika minutách dojde k odemčení souborů.

V cronu nastavíme spouštění automatických skriptů *auto_backup* a *auto_clean*. Auto_backup zajišťuje automatické zálohování nastavení a databáze, proto je vhodné ho načasovat například na půlnoc každého dne. Auto_clean odemyká zámky přístupu k souborů, pro případ, že nějakým nedopatřením nedojde k odemknutí. Jeho časování je vhodné vybrat mezi 5-10 minutami.

## 3.4 Práce s Vadminem

Po přihlášení se nám zobrazí menu. Jeho volby záleží na druhu přihlášeného uživatele.




Obr. 3.7: Přihlašovací obrazovka.


Obr. 3.8: Menu roota.

### 3.4.1 Nastavení roota

- **Domény** - editace skupin domén.
- **Soubory** - editace cest k souborům skupin domén.
- **Výrazy** - editace regulárních výrazů, pomocí nichž se v konfiguračním souboru hledají potřebné direktivy.
- **PHP** - nastavení PHP.
  - **Nastavení uživatele** - nastavení PHP jak ho vidí sám uživatel.
  - **Možnosti uživatele** - nastavení, které volby budou zvolenému uživateli přístupné.
  - **Globální možnosti** - nastavení, které volby budou globálně přístupné.
- **Profil** - změna hesla právě přihlášeného uživatele.
- **Uživatelé** - editace uživatelů.

Vyhledávání direktiv v konfiguračních souborech se provádí pomocí regularních výrazů uložených v databázi. To umožňuje libovolné změny zobrazeného nastavení.

```
^ Document root (.*)$
```

Znak ^ značí počátek řádku, pak následuje hledaná direktiva, (.*) znamená vše za hledanou direktivou, až po konec řádku $. Typ udává o jakou direktivu se jedná, zda je párová, nepárová, nebo může nebývat pouze hodnot On a Off. Podle toho pak aplikace po najití této direktivy zobrazí pole pro vložení textu nebo pouze výběr On/Off.

### 3.4.2 Tvorba skupin

Skupina se vytvoří jednoduše vytvořením souboru `conf` v adresáři *groups*. Aplikace sama zjistí tento soubor a nabídne jej u vytváření uživatelů jako možnou skupinu.

### 3.4.3 Tvorba domén

Po vytvoření uživatele program nabídne vytvoření domény. Pokud je uživatel například správce skupiny a nemá mít doménu, tuto výzvu ignorujte.

### 3.4.4 Operace s uživateli

Vadmin umožňuje standartní operace s uživateli. Vytváření nových uživatelů a správců skupin, jejich editaci a mazání.

# 4 SYSTÉM PRÁCE APLIKACE

## 4.1 Zabezpečení

Přístup do aplikace je řešen pomocí autentifikace uživatele pomocí přihlašovacího jména a hesla. Uživatelský vstup je porovnán s údaji z databáte. Jména a md5 hashe hesel uživatelů jsou uložena v tabulce „users.“ Přihlašovací stránka a skript byla testována pomocí nástrojů SQL Inject me a XSS me, s aktuálním seznamem známých SQL injection a XSS útoků. Žádný ze 153 SQL injection a 486 XSS simulovaných útoků nebyl úspěšný. Stejným způsobem bylo testováno i uživatelské rozhraní na všechny uživatelské vstupy (celkem 510 SQL a 1620 XSS testů.) Rozhraní správců skupin a roota je už nachýlnější k útoku, z důvodu více uživatelských vstupů a složitější konstrukce a hodně záleží na nastavení samotného serveru. Zde se však počítá s tím, že správci jsou sami zodpovědní. Proto je doporučeno, aby měli bezpečná hesla a často je měnili. Pro zakázání přístupu různým crawlerům stačí v *vadmin.conf* pomocí Deny from zakázat přístup konkrétní IP/doméně.

| id | login | password | active | type | group | units | php | php_settings |
|---|---|---|---|---|---|---|---|---|
| 0 | root | hash | 1 | root | all | all | 1 | |
| 1 | user | hash | 1 | user | /etc/apache2/groups/group1.conf | uzivatelova_domena | 1 | 1,2,3,4,5,7,8,10,11 |
| 2 | group | hash | 1 | group | /etc/apache2/groups/group1.conf | group | 1 | |

Obr. 4.1: Ukázka struktury tabulky users.

## 4.2 Editace souborů

Editace souborů je řešena způsobem jejich načtení do pole a testování jednotlivých řádků regulárními výrazy. To poskytuje variabilitu v hledaných direktivách. Regulární výrazy jsou uloženy v tabulce „expressions.“ Cesty k souborům jsou uloženy v tabulce „files.“ Při otevření souboru se zavolá funkce protectfile, která nastaví varování, pokud by se někdo pokoušel v průběhu editace ke stejnému souboru přistoupit. Varování se zruší při uložení souboru, nebo automaticky po zadaném časovém intervalu. To ošetřuje případ kdy uživatel otevře soubor a zavře okno prohlížeče bez uložení. Mazání domén je ponecháno na správci. V závislosti na všech možných variantách struktury hosta by automatické mazání nebylo bezpečné.

| id | expression | type | info |
|---|---|---|---|
| 0 | ^<VirtualHost (.*)>$ | 0 | Nastaveni virtualni domeny |
| 1 | ^ServerName (.*)$ | 2 | Jmeno serveru |
| 2 | ^RewriteEngine (.*)$ | 1 | Rewrite mod |
| 3 | ^VirtualDocumentRoot (.*)$ | 2 | Root |
| 4 | ^ServerRoot(.*)$ | 2 | Root serveru |
| 5 | ^safe_mode_gid(.*)$ | 1 | popis |
| 6 | ^DocumentRoot (.*)$ | 2 | Root serveru |

Obr. 4.2: Ukázka struktury tabulky expressions.

## 4.3 Nastavení

Nastavení konstant je uloženo v souboru *include/config.php*.

```
define("SQL_HOST","localhost");
define("SQL_DBNAME","vadmin");
define("SQL_USERNAME","uzivatel");
define("SQL_PASSWORD","heslo");

define("BACKUP","/var/www/backup/");
define("GROUPS","/etc/apache2/groups/");
define("DAYS","2");
define("SECURITY","/var/www/security/");
define("MINUTES","7");
```

V první části jsou přihlašovací údaje k SQL databázi. BACKUP - cesta k adresáři pro automatickou zálohu. GROUPS - cesta k adresáři s konfigurací skupin. DAYS - kolik dní se archivují zálohy. SECURITY - cesta k adresáři, kam se ukládají zámky souborů. MINUTES - po kolika minutách dojde k odemčení souborů.

# 5 ZÁVĚR

Z důvodu neexistence vhodného webového rozhraní pro konfiguraci apache serveru a především virtuálních hostů, byla vyvinuta vlastní aplikace. Při jejím návrhu se musel najít kompromis mezi požadavky administrátora a uživatele. Administrátor vyžaduje co nejširší možnosti nastavení a variabilitu. Uživatel klade nároky převážně na přehlednost a ergonomii ovládání. Proto aplikace obsahuje více druhů uživatelů. Menu se pak mění podle předpokládaných znalostí uživatele. Root má možnost nastavit celý server, protože je u něj předpokládána nejhlubší znalost problematiky. Zde také aplikace umožňuje velikou variabilitu nastavení pomocí regulárních výrazů. Správce skupin se nemusí starat o nastavení serveru, stará se pouze o uživatele ve své skupině. Na nejnižší úrovni je pak běžný uživatel, na kterého nejsou kladeny žádné nároky. Má možnost měnit volby PHP pro svou doménu (pokud mu to je správci dovoleno.) Aplikace byla pojmenována Vadmin, protože jejím primárním úkolem je vytváření a správa virtuálních hostů. Byla vytvořena pomocí masivně používaných technologií s volnou licenční politikou ( Apache2, PHP, MySQL, Debian.) To zajišťuje, že aplikaci bude možné použít na širokém rozsahu existujících serverů.

# LITERATURA

[1] ZAJÍC, Petr. *Seriál PHP* [online]. 2004 , 27.5.2004 [cit. 2007-12-14]. Dostupný z WWW: <http://www.linuxsoft.cz/php/>.

[2] The Apache Software Foundation. *Apache HTTP Server Version 2.2 Documentation* [online]. c2007 [cit. 2007-11-10]. Angličtina. Dostupný z WWW: <http://httpd.apache.org/docs/2.2/>.

[3] VEERLE, Pieters. *Verlee´s blog : A CSS styled table* [online]. 2005,25.6.2005 [cit. 2007-11-13]. Dostupný z WWW: <http://veerle.duoh.com/blog/comments/a_css_styled_table/>.

[4] The PHP Group. *PHP: Hypertext Preprocessor* [online]. c2007 , Sat Dec 15 10:45:27 2007 PST [cit. 2007-12-15]. Dostupný z WWW: <http://www.php.net/>.

# SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

KDE K Desktop Environment

PHP Hypertext Preprocessor

SQL Structured Query Language

GUI Graphical User Interface

SDL Storage Definition Language

VDL View Definition Language

DDL Data Definition Language

DML Data Modification Language

DDL Data Control Language

SQL injection Propašování speciálního kódu do uživatelského vstupu, za účelem provedení vlastního SQL požadavku.

XSS Cross-site scripting, podstrčení vlastního javascriptového kódu do stránek.

CRAWLER Internetový robot procházející internet a většinou indexující obsah.

SYMLINK Speciální soubor obsahující cestu k souboru, na který odkazuje.

MD5 Hashovací funkce pro vytváření otisků (hashů), pevné délky

CRON Unixový daemon, zajišťující spouštění úloh v daný čas.

# SEZNAM PŘÍLOH

# A PRVNÍ PŘÍLOHA

CD s aplikací vadmin,návodem na obsluhu a elektronickou verzí této práce.